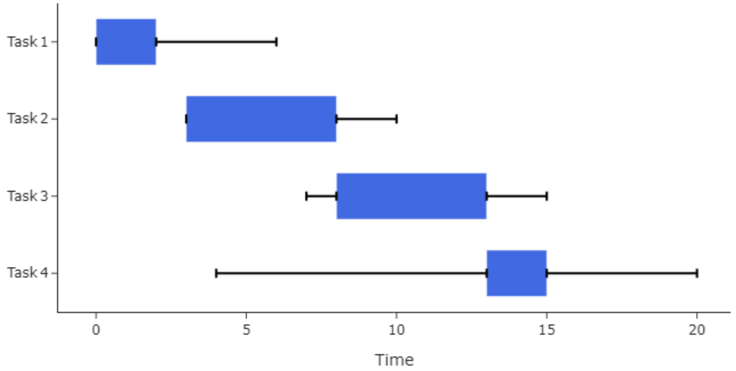
Explaining detectable precedences for the disjunctive constraint

Matthias van Vliet (m.b.vanvliet@tudelft.nl) | Supervisors: Emir Demirović & Imko Marijnissen

1. Introduction

- **Constraint Programming (CP):** paradigm for solving combinatorial problems by defining the constraints a solution must satisfy
- **Propagator:** a function which filters variable domains according to a constraint
- **Lazy clause generation (LCG):** CP solving technique where propagations need to be explained to enable techniques such as backjumping
- **Disjunctive:** a constraint which does not allow any two tasks to overlap
- **Detectable precedences (DP):** a propagation rule for the disjunctive where task i must precede task j if the earliest completion time of j is greater than the latest starting time of i

1. Example situation to illustrate propagation using DP



$$S_2 \geq 3 \land S_2 \leq 5 \land S_3 \geq 7 \implies S_3 \geq 8$$
$$S_2 \geq 3 \land S_2 \leq 5 \land S_4 \geq 4 \implies S_4 \geq 8$$

Explanations for Propagations using DP in the example

- **Gap:** adapting DP propagation algorithm by Fahimi et al. [1] to support explanations in an LCG solver and benchmark the influence of the explanations on performance metrics

2. Approach

Three approaches to generating explanations described and implemented:

1. **Naïve:** Conjunction of all bounds of all starting times
2. **Previously scheduled (novel intermediate):** Only include preceding tasks in the explanation
3. **Last cluster (novel advanced):** Only include the set of contiguously scheduled tasks that 'push' the propagated task and then 'lift' the explanation.

3. Method

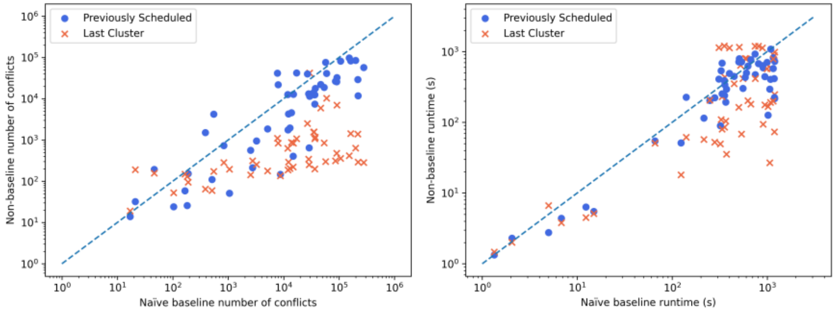
- Implemented DP propagation algorithm by Fahimi et al. [1] in the Pumpkin solver
- Implemented Naïve, Previously scheduled and Last cluster explanations
- Benchmarked the three approaches on 50 jobshop instances

4. Results

- Avg. #conflicts 43% of naïve for previously scheduled and 4% for last cluster
- Last cluster appears to have difficulty with large instances when solving is not close to optimality
- Decomposition is on average 18x faster than Last cluster, but struggled to prove optimality

	Naïve (baseline)	Previously scheduled	Last cluster
Avg. #conflicts	45K	19K	1.8K
Avg. LBD	21.31	19.09	8.62
Avg. runtime ratio with baseline	1	0.822	0.773

2. Aggregate metrics of the three approaches



3. Non-baseline approaches plotted against baseline naïve approach

5. Conclusion

- Previously scheduled better than baseline and last cluster noticeably better than baseline across all metrics
- Investigate Last cluster runtime difficulty with large instances further
- Combine Last cluster with other propagation rules to see whether it can compete with decomposition

References

[1] H. Fahimi and C. Quimper. Linear-time filtering algorithms for the disjunctive constraint (2014)